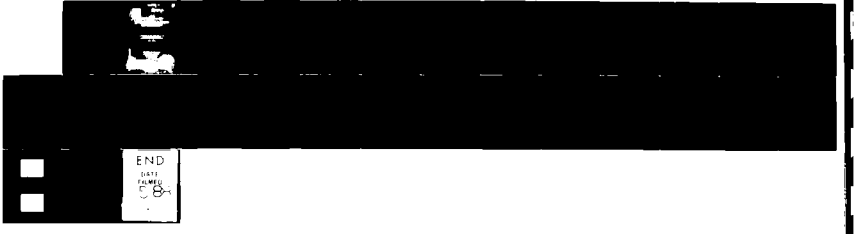
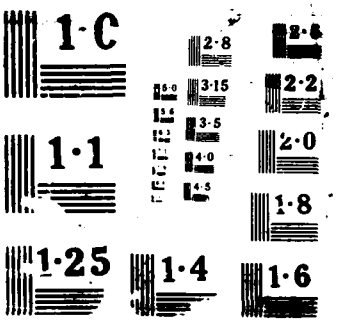
AD-A191 852 AN INTERFACE FOR THE CONNECTION OF THE SCP2160A PULSE ANALYSER TO A HEWLE..(U) ROYAL SIGNALS AND RADAR ESTABLISHMENT MALVERN (ENGLAND) C WILLIAMS 09 OCT 87 1/1

UNCLASSIFIED RSRE-MEMO-4030 DRIC-BR-104726 F/G 14/2 NL




1·C
1·1
1·25
1·4
1·6
2·8
3·15
3·5
4·0
4·5
2·2
2·0
1·8

ROYAL SIGNALS AND RADAR ESTABLISHMENT

Memorandum 4030

TITLE: AN INTERFACE FOR THE CONNECTION OF THE SCP-2160A PULSE ANALYSER TO A HEWLETT-PACKARD DESK-TOP COMPUTER

AUTHOR: C. Williams.

DATE: 9 October 1987

SUMMARY

This memorandum describes a digital First-In-First-Out Memory (FIFOM), with control circuitry, which is designed as an interface between the Marlborough Communications SCP-2160A Pulse Analyser and any desk-top computer fitted with a H-P 98622A GPIO Interface. The FIFOM Interface is necessary to accommodate the data transfer-speed differential that exists between the SCP-2160A and the GPIO port of the computer.

The SCP-2160A is a 160 MHz I.F. pulse processor that digitises, displays, and stores the pulse parameters that are required for ESM applications.

CONTENTS

ILLUSTRATIONS

1. Introduction

The SCP-2160A has an internal memory store for the descriptors of 256 pulses. This is insufficient for carrying out meaningful statistical analysis on the received pulse trains. Data on every pulse within the pulse amplitude acceptance range, and the 25 MHz band-width of the SCP-2160A, are available at the Direct Memory Access (DMA) output port on the rear panel of this equipment. This data is passed at approximately four times the speed at which the GPIO part of the computer will successfully accept data. The FIFOM Interface is, therefore, used as a buffer store to hold pulse data until the GPIO is able to accept further data input.

2. Pulse Analyser DMA Data Output

The SCP-2160A represents the current state of the art in I.F. pulse processing. This entails digitising, displaying and storing, on an individual pulse basis, not only the pulse width and the time of intercept (TOI), but also the I.F. frequency to 100 KHz resolution, and the amplitude. This information is encoded in five 16 bit words (See Fig. 1), following the scheme outlined in Fig. 2.

Included in the DMA output port with the pulse data, are three other outputs that are used by the FIFOM interface control circuitry. The first of these is the Threshold Event pulse which is active only if the initiating I.F. pulse is longer than the analyser's short pulse trap time and if the pulse amplitude exceeds the operator-set threshold. The Threshold Event pulse indicates that pulse-parameter data is to follow. The Data Strobe is the second of the control outputs, and is a train of five pulses that clocks the five 16-bit descriptors out of the SCP-2160A and into the FIFOM. The third of the control outputs is the qualification, or Qual., signal. This pulse indicates that the following data block corresponds to a pulse contained within the limits of the operator-set Qual frequency/amplitude box on the pulse analyser CRT display.

A simplified timing diagram of the digitisation and qualification of one input I.F. pulse is shown in Fig. 3. Waveform 1 is an illustration of a pulse-modulated sine wave signal in the presence of some noise. This signal provides the log video signal, waveform 2, and the frequency modulated video signal shown in waveform 3. The operator-set threshold level is shown on waveform 2. When the log video pulse rises through this threshold value, a TTL Threshold Event pulse is generated as shown in waveform 4.

The digitised amplitude and frequency values are compared with the operator-set limits of the Qual box on the CRT. If these values fall within the limits, then the signal is Qualified and a TTL level pulse is generated, as shown by waveform 5.

The log video signal is delayed by 400 nsec so that timing measurements may be carried out on Qualified pulses only. Values for Time of Intercept and pulse width are then generated, as per waveform 6.

DMA data from the SCP-2160A are available from a 55-way connector on the back panel of the equipment. Each data and control line is provided as a differential output from a line driver. Data describing each pulse is composed of five words each being sixteen bits wide. See Fig. 4 for the output assignments.

3. Computer GPIO Interface

The HP98622A GPIO Interface is used in preference to the General Purpose Interface Bus because it is faster in operation, and allows the manipulation of 16-bit data transfers between the FIFOM Interface and the computer. The sixteen data input lines are divided into two 8-bit bytes. The GPIO is capable of transferring 930,000 16-bit words per second.

Four peripheral control and status lines are utilised, namely, the Extended Control Output Line ($\overline{\text{CTL1}}$), the Peripheral Control Line (PCTL), the Peripheral Flag Line (PFLG), and the Extended Status Input Line ($\overline{\text{STI0}}$). $\overline{\text{CTL1}}$ is a latched line that is used as an input to the data transfer control circuit in the FIFOM Interface to control the transfer of the pulse analyser output data. PCTL is an outgoing line to the FIFOM Interface, and is used to shift data out of the FIFOM and into the GPIO. PFLG is controlled by the FIFOM Interface, a logic high state on this line indicating that the interface has data for transfer. This line is paired with PCTL to synchronise the Computer/FIFOM handshaking. $\overline{\text{STI0}}$ is used to monitor the state of the FIFOM composite Input Ready flag. The latter is sampled a number of times by the computer. If four successive samples each indicate that IR is not high, then the FIFOM is assumed to be full.

Before use, the GPIO Interface board must be correctly configured, by setting each of the four interface-configure switches to the required states. These switches allow the user to select the data-in clock source, the logic sense of the data and handshake lines and to select full-mode or pulse-mode handshaking, and to set the interface select code, and the peripheral interrupt priority.

The Data-In Clock Source switch allows each 8-bit byte to be clocked in by its own clock source, although in this application, both bytes are clocked from the same source, BSY. Here, a transition on the PFLG line from READY (RDY) to BUSY (BSY), from high to low in this application, clocks data from the FIFOM Interface into the input registers of the GPIO Interface.

Using the Option Select Switch, the logic sense of the data-in lines (DIN), the handshake (HSHK); PCTL, and PFLG were set as follows:-

DIN	LOW = 0	HIGH = 1
DOUT	NOT USED	
HSHK	FULL	
PSTS	NOT USED	
PCTL	LOW = CLEAR	HIGH = SET
PFLG	LOW = BUSY	HIGH = READY

All GPIO Interface boards are set to select code 12 as despatched, and this was left unaltered.

The peripheral priority interrupt level switch was also unaltered, since the SCP-2160A is the only device using the GPIO. The PCTL delay was also unaltered.

The Data Out (DOUT) jumper was installed, ensuring that, at power up and after an interface reset, both upper and lower data-byte registers are cleared.

In order to allow DMA operation, the Burst jumper was removed, thereby enabling the Direct Memory Access Controller (DMAC) to hold the System bus for a short period of time following each DMA transfer to the GPIO Interface. If the peripheral completes the data handshake during this period, another DMA transfer can occur without the DMAC having to re-acquire the the system bus. This results in a higher data transfer rate.

An interface circuit is required at the input to the GPIO. The data output from the FIFOM Interface consists of sixteen twisted-pairs, whereas the GPIO data input is via sixteen single lines. This GPIO input interface consists of the necessary permanently-enabled line receivers. These are powered by the FIFOM Interface power supply. The GPIO GND connectors are joined with the FIFOM Interface ground.

The GPIO pin assignment is provided in Fig. 5.

4. Outline of FIFOM Interface design

For the purposes of description, the Interface may be divided into six sub-sections, namely, the data transfer control circuit, the pulse qualification-source circuit, the Data Strobe timing cirucit, the data input circuit, the FIFOM, and the data output circuit.

i) Data Transfer Control Circuit

The operation of this circuit may be more readily understood if reference is made to the FIFOM Interface timing diagram in Fig. 6, and the circuit diagram in Fig. 11.

This circuit allows the computer to control the flow of data into the FIFOM, by providing one of the two inputs required by the AND gate, G2, for controlling the Data Strobe line receiver enable signal.

Primary control of data transfer is effected by the Extended Control Output Line, $\overline{CTL1}$, from the GPIO Interface. However, precautions have to be taken to ensure that $\overline{CTL1}$ does not initiate transfer in the middle of a 5-word pulse description. Such a condition would cause partial transmission of the pulse descriptor word and would make synchronisation of subsequent data words difficult.

When brought on to the FIFOM Interface board, this line is biased by the recommended resistive-divider receiver circuit, and is then used to switch monostable, M3, and gate G1. Once M3 is set, the state of the Q output is independent of further transitions of the A and B inputs, until $\overline{CTL1}$ goes low, clearing the output. A positive-going transition on the B input sets M3, provided that AND gate, G1, is not inhibited by a blanking pulse derived from the Threshold Event Signal.

The output from the Threshold Event line receiver is a positive-going 700 nsec pulse. This is stretched to a duration of 1.3 usec by a re-triggerable monostable, M1, so as to place the trailing edge of this pulse in front of the leading edge of the first of the Data Strobe pulses. The trailing edge of this stretched pulse is then used to trigger monostable M2 to produce a negative-going pulse that must be long enough, of the order of 2.2 usec, to completely envelope the five pulses of the Data Strobe pulse train. This provides the blanking pulse which prevents the start of data transfer during a 5-word descriptor block, by inhibiting G1. This pulse generation and stretching is necessary to overcome timing errors in the SCP-2160A and should be carried out with regard to the equipment being used.

If $\overline{CTL1}$ goes high during the Data Strobe pulses period, the output of G1 will remain low, since the blanking pulse inhibits this gate. Therefore, M3 will produce no output signal. Since $\overline{CTL1}$ is latched, it will remain in the high state, and when the blanking

pulse terminates, both inputs to G1 will be high, thereby triggering M3 to produce a Data Strobe enable signal.

ii) Pulse Qualification-Source Circuit

This circuit allows the operator the choice of accepting data on all pulses within the pass-band of the SCP-2160A, or only of those pulses contained within the operator-set Qualification box on the analyser display. Accepting all pulses entails placing a +5V signal on the appropriate input to G2. The selection of Qual utilises the Qual active-high signal that is available as a co-axial output on the rear panel of the SCP-2160A. Because of timing errors in the analyser it was found necessary, using a tapped delay line, D1, to delay the Qual pulse on the FIFOM Interface board so that it fully envelopes the five pulses of the Data Strobe pulse train, as shown in Fig. 7, and avoids data corruption.

A logic high state produced by the Qual source, together with a high output from M3 in the data transfer control circuit, places a logic high on the output of G2, thereby enabling the Data Strobe line receiver.

iii) Data Strobe Timing Circuit

Timing errors in the SCP-2160A necessitate the introduction of a delay line, D2, to ensure the correct positioning of the Data Strobe pulses with respect to the pulse descriptors that they are clocking.

The Data Strobe Signal is available as a differential output from the SCP-2160A, and its twisted-pair output is terminated, on the FIFOM Interface board in 110 Ω and fed to a line receiver. This line receiver has two ENABLE lines, one being active high and the other active low. The active low Enable, ENABLE is taken to the +5V supply rail, thereby providing only one, active-high, Enable input. The ouptut of G2 provides this signal if the gate inputs are both high.

A 1 KΩ resistor terminates the output from the line receiver to ensure that the Shift-In line to the FIFOM is held low when the Data Strobe line receiver is disabled. This prevents the input of spurious data. The delay line is required to position the leading edges of the Data Strobe pulses in the centres of the pulse-parameter data words. It is the leading edges of the Data Strobe pulses that clock the data. Such positioning ensures that the pulse data has settled after any transition and allows a small margin in relative Data Strobe/pulse data positioning. This is shown in Fig. 8.

The Data Strobe line receiver has insufficient output current drive to supply a resistor low enough in value to hold down the delay line input when necessary, so that a NAND gate buffer is included to provided this drive. A NAND gate is provided at the

delay line output to re-invert the pulse train and to restore true TTL levels. NAND gates were chosen to provide shorter propagation times than with other TTL circuits and to allow greater versatility in gate provision.

iv) Data Input Circuit

The data input circuit consists solely of line receivers and terminating resistors for the sixteen twisted-pair pulse data inputs. Each line receiver is hard-wired in the enabled state, since data transfer is controlled by the Data Strobe circuit.

v) FIFOM

The FIFOM used in this application is the TRW Inc. TDC 1030 which has its memory organised as 64 words by 9 bits. It is capable of a 15 MHz shift-in shift-out data rate when used with the control flags. The FIFOM is expanded to accept 16-bit data by using the devices in a parallel arrangement, eight bits from each FIFOM being used. To minimise problems caused by the memory filling up under high pulse repetition rate conditions, eight FIFOMs are used in a four-pairs combination to provide 256 words by 16-bits wide. The first stage Input Ready flags and the final state Output Ready flags are ANDed to provide composite control flags. Output Enable, $\overline{OE}$, is wired low to ensure that the outputs are TTL compatible, Master Reset, $\overline{MR}$, is taken to a panel-mounted momentary action push-button switch.

Following power up, the Master Reset line is automatically pulsed low to clear the FIFOM. The Input Ready (IR) flag high indicates that the FIFO input stage is empty and is available to receive data. When IR is valid, Shift In (SI) may be asserted, thus loading the data present at the FIFOM inputs into the input stage. Bringing the SI signal high, causes IR to drop low. With SI low, the data in the input stage propagates asynchronously through the FIFOM to the output stage if the FIFOM is empty, or to the last empty locations. If the FIFOM is not full after the SI pulse, IR will again be valid, indicating that there are empty locations available for data. The IR flag remains low if the memory is full. See Fig. 9.

The Output Ready (OR) flag high indicates that there is valid data in the output stage. When the OR flag is valid, data can be transferred out of the FIFOM by the Shift Out (SO) control. A SO high state results in a busy (low) signal at the OR flag. When SO is brought low, data is shifted out of the FIFOM. At the completion of the SO pulse OR goes high. If the last valid data has been shifted out, the OR flag remains low but the last word remains on the output pins.

With the FIFOM empty, apart from this last word, SO can be held high until a Shift-In pulse is generated. Following SI, the data falls through to the output stage. This results in an OR high and data is shifted out. SO must be brought low before additional data can be shifted out. See Fig. 10.

A problem arises with the data word remaining in the FIFOM at the end of the Shift-Out sequence. It is not practical simply to clear the FIFOM before initiating data input because the software transfer statement requires a full buffer to be present at the start of the transfer sequence. The controlling software is described in a subsequent section.

vi) Data Output Circuit

The pulse data output circuit consists of four differential line drivers for the sixteen twisted-pair output lines. Each is hard-wired in the enabled state. Data output is controlled by the PCTL line from the GPIO port. This line biased by the required resistive - divider receiver circuit.

5. Control Software

This section describes a sub-program "Panalyser" that is used to set up, execute, and control, the data transfers between the pulse analyser and the computer. It should be noted that this sub-program will not run as a separate entity without modification. The original was called from a mother program. A listing of this sub-program is given on page 12.

DMA transfers are executed to a pre-established buffer in the computer by an overlapped transfer, that is, input to, and output from, the buffers, are carried out simultaneously. Other tasks may also be carried out concurrently by the processor.

The COM statements in lines 70 and 80 are used to dimension, and reserve, memory for variables in a special common memory area so that these variables can be accessed by the sub-program and by the calling program. The data path specifiers are included in COM to enable the sub-program to be exited whilst transfers using these paths are in progress. Pbuffer$ BUFFER is used as a byte buffer and must be dimensioned in the program that calls "Panalyser". The size of this buffer determines the number of pulses that can be stored. Each pulse is described by five words, each consisting of two 8-bit bytes. Therefore, each pulse is described by ten bytes. Pbuffer$ BUFFER must, therefore, be dimensioned as the number of pulses required X 10.

"Panalyser" has three distinct and separate modes of operation that are controlled by the value passed into the "Flag" parameter, namely:-

Flag = 2 Set up pulse analyser

Flag = 1 Start transfer

Flag = 0 Stop transfer and store data

The three cases are described here:-

CASE 2 Lines 120 - 140 set up the received pulse analyser signal with reference to the noise level. This is specific to the output level of the receiver used in conjunction with the pulse analyser.

Lines 150 - 190 set up the pulse analyser in the mode required for use on the General Purpose Interface Bus (GPIB). This is not relevant to DMA transfers but is required to allow the pulse analyser to be used simultaneously on both GPIB and DMA ports.

Pa-address is the GPIB address of the pulse analyser and consists of the Interface Select Code (7 for the GPIB) and the primary address used to specify the analyser, (16 in this case). Therefore Pa-address equates to 71600. The trailing zeroes form the secondary address and specify the internal functions of the pulse analyser. For example, line 120 is thus interpreted as:-

OUTPUT 71603 ; "1"

This statement loads a "1" into address 71603 to select Noise-Riding Threshold (N.R.T.).

CASE 1 Paths @Pbuff and @Pfile will be assigned if Case 0 has already been called. Lines 220 and 230 close these paths, and would cause the program to wait if previous data filing were not complete.

The GPIO reset in line 250 ensures that the GPIO is in a known state before data transfer is started.

A problem exists with the data transfers in that they take time to set up and cannot be set up from an empty source, the FIFOM in this case. The solution is to fill the FIFOM with any data, disable the inputs via $\overline{CTL1}$, and then to start the transfer. When the FIFOM is empty, (apart from the last data word - see section (v) in the Outline of FIFOM Interface design), the FIFOM inputs are enabled, allowing data to run through into the computer buffer. This transfer is terminated after a delay specified by the WAIT

instruction, which ensures that the FIFOM is empty. The FIFOM inputs are then enabled in anticipation of further data.

The compound Input Ready (IR) flag in the FIFOM is sampled asynchronously four times by the loop in lines 270 to 330, until none of the four samples indicate that IR is high. This allows distinction to be made between the "busy" low state and the FIFOM full state (see Fig. 9). Once the FIFOM is full, a data transfer may be initiated.

CASE 0 Line 410 terminates the transfer if this is required before the computer buffer is full, (the transfer stops automatically when the buffer is full). This termination could be required, for example, if an emitter closed down during data collection, before the buffer was full.

The statement in line 430 sets the buffer status register empty pointer to the number of the first data byte to be accepted. In the case of one FIFOM pair (16 bits), the required figure is 129, i.e. (2 x 64 bytes) + 1.

In the case of a four-pair FIFOM, such as is used in the final development interface, the required figure is

4 (2 x 64) + 1 = 513

The statement in line 430 has the effect of throwing away any spurious data that might be contained in the first FIFOM - full of information. Refer to the final paragraph of the description of the FIFOM in section (v).

Status Register 4, referred to in line 440, provides the number of data bytes in the buffer and places the value in the variable, Nob.

Line 450 creates a binary data file slightly larger than is required to ensure that all of the data bytes are stored successfully.

Line 470 transfers the data bytes and counts them to ensure that a whole number of bytes is transferred. The sub-program can be exited once this transfer is initiated, since it will automatically terminate transfer when the byte count is satisfied.

```
10    ! "Panalyser" CONTAINS SUB Panalyser
20    ! NCWF/MGH 16/09/86
30    SUB Panalyser(INTEGER Flag)
40    ! Flag=2 SET-UP;Flag=1 START GPIO TRANSFER
50    ! Flag=0 TERMINATE & STORE DATA
60      OPTION BASE 1
70      COM /Panalyser/ @Pdevice,@Pbuff,@Pfile,Pbuffer$ BUFFER
80      COM /Pulse_anal_addr/ Pa_address
90      INTEGER Ip_ready,I,Stat_ip
100     SELECT Flag
110     CASE 2                                          ! SET-UP
120       OUTPUT Pa_address+3;"1"                       ! N.R.T
130       OUTPUT Pa_address+8;"12"                      ! ATTEN=12dB
140       OUTPUT Pa_address+9;"15"                      ! 15dB THRESHOLD
150       OUTPUT Pa_address+4;"0"                       ! PW ACCEPT
160       OUTPUT Pa_address+16;"002"                    ! MIN PW=.2uSec
170       OUTPUT Pa_address+17;"200"                    ! MAX PW=20uSec
180       OUTPUT Pa_address+6;"1"                       ! MEM MODE
190       OUTPUT Pa_address+5;"1"                       ! PRI MODE
200       OUTPUT Pa_address+14;"2"                      ! INTERNAL TOI RESET
210     CASE 1                                          ! START TRANSFER
220       ASSIGN @Pbuff TO *                            ! ACTIVE AFTER CASE 0
230       ASSIGN @Pfile TO *                            ! ACTIVE AFTER CASE 0
240       OUTPUT Pa_address+14;"0"                      ! SET EXTERNAL TOI RESET
250       CONTROL 12;1                                  ! RESET GPIO INTERFACE
260       CONTROL 12,2;0                                ! ENABLE INPUTS TO FIFO
270       REPEAT                                        ! UNTIL I/P READY ALWAYS LOW
280         Ip_ready=0
290         FOR I=1 TO 4                                ! LOOK AT I/P READY 'I' TIMES
300           STATUS 12,5;Stat_ip
310           IF BIT(Stat_ip,0)=0 THEN Ip_ready=1
320         NEXT I
330       UNTIL NOT Ip_ready                            ! FIFO FULL
340       CONTROL 12,2;2                                ! DISABLE FIFO INPUTS
350       ASSIGN @Pbuff TO BUFFER Pbuffer$
360       ASSIGN @Pdevice TO 12;WORD                    ! GPIO
370       TRANSFER @Pdevice TO @Pbuff                   ! START TRANSFER
380       WAIT .001                                     ! ENSURE FIFO EMPTIED
390       WRITEIO 12,7;0                                ! ENABLE FIFO INPUTS
400     CASE 0                                          ! TERMINATE TRANSFER
410       ABORTIO @Pdevice                              ! STOPS TRANSFER (IF NOT DONE)
420       ASSIGN @Pdevice TO *
430         CONTROL @Pbuff,5;129                        ! IGNORE FIRST 128 BYTES
440         STATUS @Pbuff,4;Nob                         ! NUMBER OF BYTES
450           CREATE BDAT Filename$,(Nob/256)+2
460           ASSIGN @Pfile TO Filename$
470           TRANSFER @Pbuff TO @Pfile;COUNT INT(Nob/10)*10
480       OUTPUT Pa_address+14;"2"                      ! INTERNAL TOI RESET
490     END SELECT
500     SUBEXIT
```

6. Setting up the Interface

Setting up the Interface initially consists of correctly positioning the blanking pulse with respect to the data strobe pulse train, as referred to in 4 (i), (see also Fig. 6). Positioning is effected by varying R1 and R2 (see Fig. 11) as appropriate.

The positioning of the Qual pulse in relation to the Data Strobe pulse train is effected by adjusting the delay as appropriate, as shown in Fig. 7.

Positioning the Data Strobe pulse train correctly in relation to the pulse data from the SCP-2160A is vital if pulse data corruption is to be avoided. Timing is effected by the delay line, D2, but the output from G4 must be monitored to ensure that this delay is correct. Refer to Fig. 11.

The test equipment used to set up the Interface was a Tektronix 7603 oscilloscope using a 7D01 logic analyser plug-in together with a DF1 display formatter used in timing diagram mode. The "least significant" line of the pod monitoring the less-significant 8-bit byte was used as the Data Strobe monitor.

Data monitoring can be conveniently carried out by considering the first pulse of the Data Strobe, which clocks in the frequency and amplitude descriptors, and the second pulse which clocks in the pulse width descriptor.

By setting up a pulse analyser I.F. input signal of known parameters, the logic analyser cursor may then be set at the leading edge of the first pulse of the Data Strobe pulse train (see Fig. 12). The cursor read-out at the bottom of the logic analyser display may be interpreted by reference to the encoding scheme detailed in Fig. 2.

For example, the frequency and amplitude data are decoded thus:-

Frequency = [(M.S. Byte of Word 1) - 128] / 10

Amplitude = [(L.S. Byte of Word 1) - 128] / 3 - 35

Moving to the second Data Strobe pulse enables the pulse width data to be decoded from the four BCD decodes. See Fig. 13.

Correct interpretation of these two descriptor words was inferred to be sufficient indication that the interface was functioning correctly. Complete testing was carried out by running an appropriate computer program to display all of the pulse train parameters.

7. Performance of the Interface

The FIFOM Interface is capable of transferring data at a mean rate corresponding to 186,000 pulses per second, which is the maximum throughput limit imposed by the computer GPIO in its burst DMA mode. The DMA pulse throughput period of the SCP-2160A is approximately 8 microseconds. This translates into a frequency of 125,000 pulses per second, which the FIFOM Interface will accept until it is full and data corruption occurs.

The peak throughput limit may present a problem when a high-density, high-PRF environment is being sampled. One solution to this may be to count in the bytes to the FIFOM and then count them out into the GPIO Interface, using byte counters. However, this possibility has yet to be addressed.

Timing errors in the Data Strobe / pulse data positioning and in the Qual pulse / Data Strobe positioning, were present in the SCP-2160A that was used in the initial setting up of the FIFOM Interface. This may have been a problem with the equipment used. An SCP-2160A borrowed for a trial period required no further alteration to the timings incorporated into the FIFOM Interface.

8. Conclusions

An interface has been designed and constructed that allows the pulse descriptor data stream from the SCP-2160A Pulse Analyser to be transferred to a Hewlett-Packard desk-top computer. This interface is capable of transferring approximately 120,000 pulse per second within the 25 MHz bandwidth of the SCP-2160A.

9. Acknowledgements

This memorandum would not have been possible without the assistance and advice that was freely given by my colleagues Nick Fane and Martin Hutchinson of the Non-Communications E.W. section of BS3 Division, and by Richard Padley. Nick Fane also wrote the controlling software. The co-operation of the aforementioned is gratefully acknowledged.

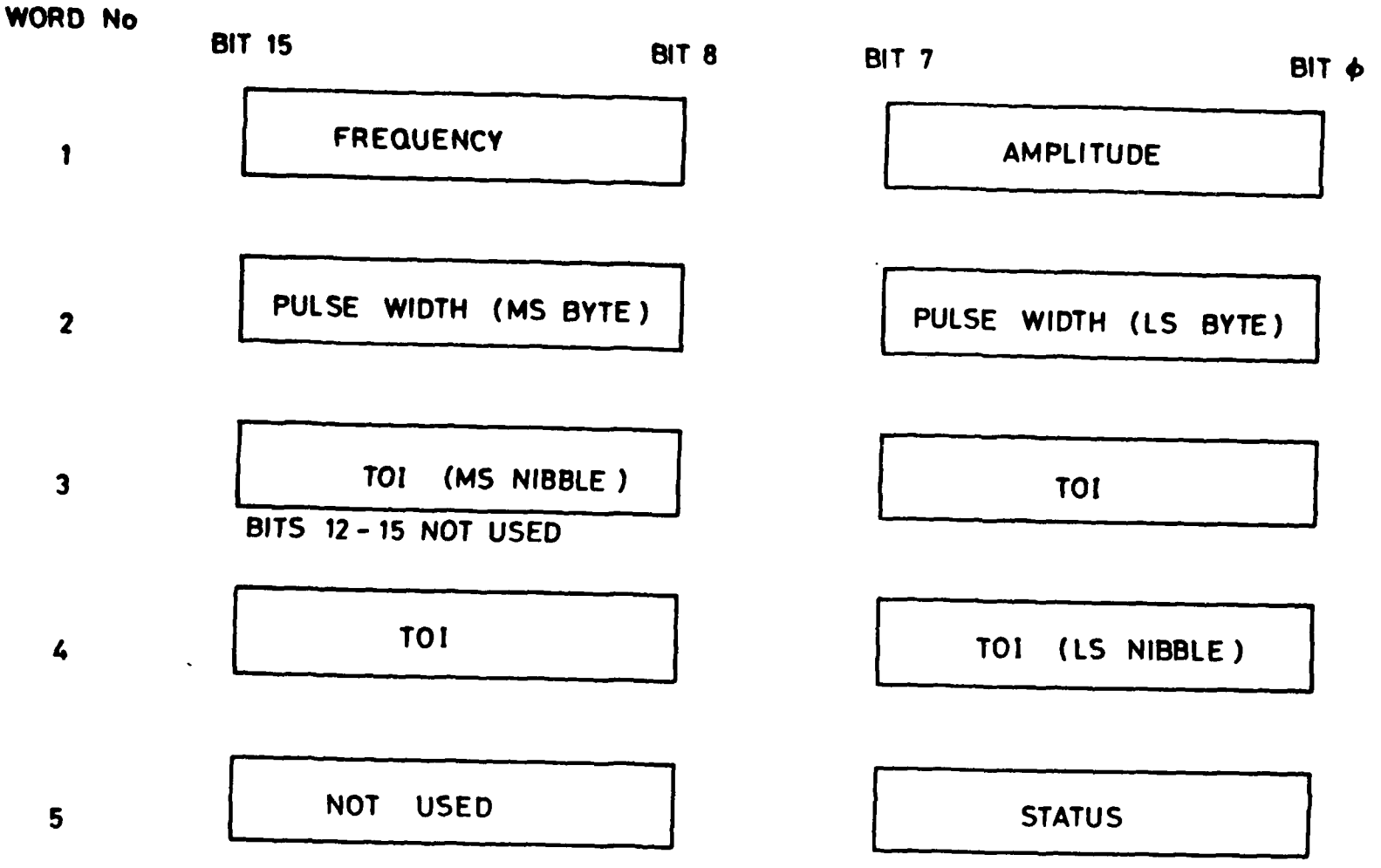


FIG.1 PULSE DESCRIPTOR WORD FORMAT

FREQUENCY (8 bits)

INTEGER (10 * FREQUENCY + 128)

AMPLITUDE (8 bits)

INTEGER [3 * (AMPLITUDE + 35) + 128]

PULSE WIDTH (16 bits)

(PULSE WIDTH * 100) encoded as 4 BCD decades

TIME OF INTERCEPT (TOI) (24 bits)

(TOI * 100) encoded as 6 BCD decades

STATUS (8 bits)

State of each bit to be interpreted separately

FIG. 2 DMA DATA ENCODING SCHEME

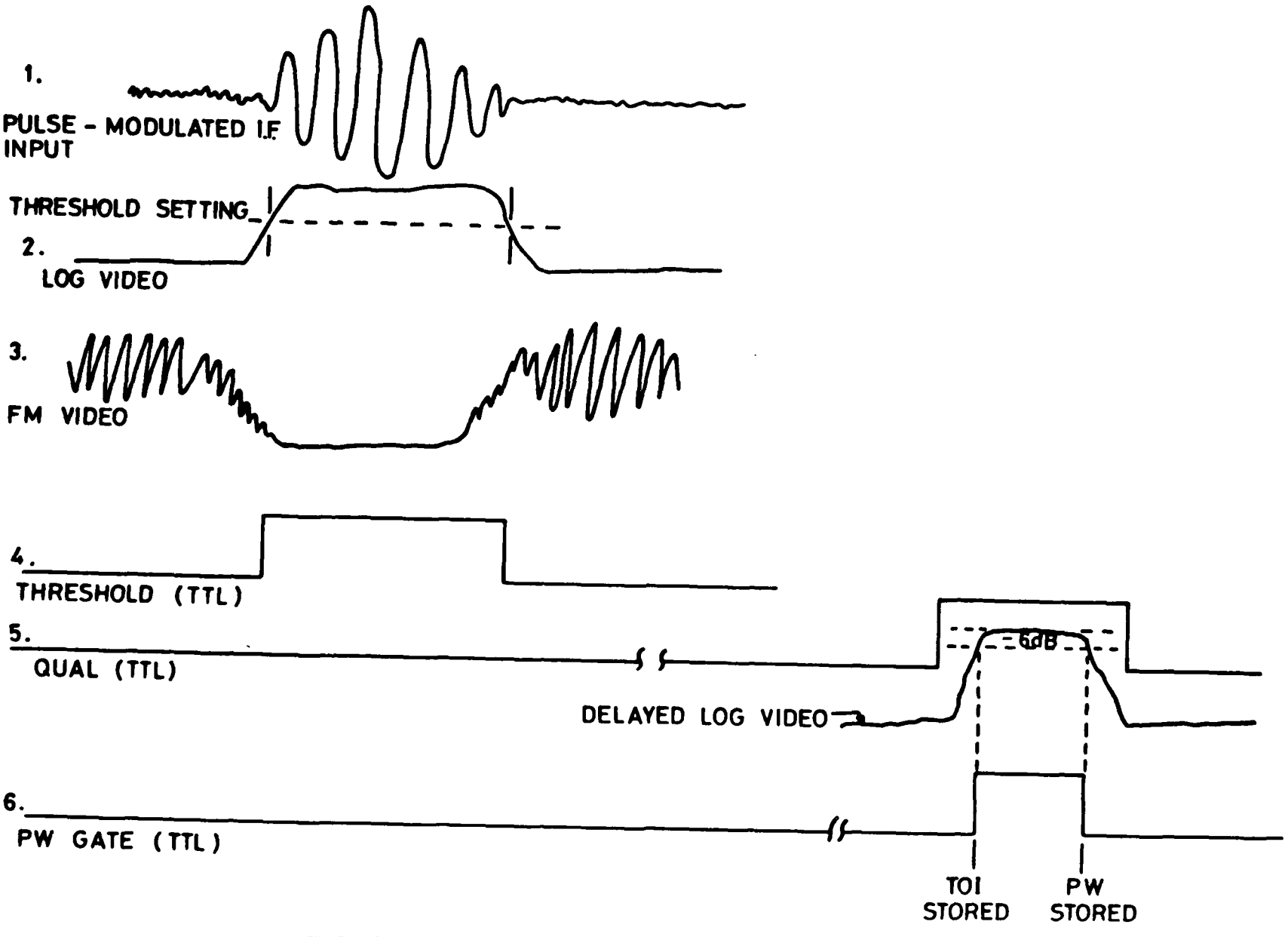


FIG. 3 SIMPLIFIED WAVEFORM TIMING

FIG. 4. SCP-2160A DMA OUTPUT ASSIGNMENT

Pin Designation	Signal Reference Designation
A	DO+
B	DO-
C	D1+
D	D1-
E	D2+
F	D2-
G	D3+
H	D3-
J	D4+
K	D4-
L	D5+
M	D5-
N	D6+
P	D6-
R	D7+
S	D7-
T	D8+
U	D8-
V	D9+
W	D9-
X	D10+
Y	D10-
Z	D11+
a	D11-
b	D12+
c	D12-
d	D13+
e	D13-
f	D14+
g	D14-
h	D15+
i	D15-
r	DATA STROBE+
s	DATA STROBE-
t	DMA START+
u	DMA START-
v	THRESH EVENT+
w	THRESH EVENT-
BB	OV
CC	OV

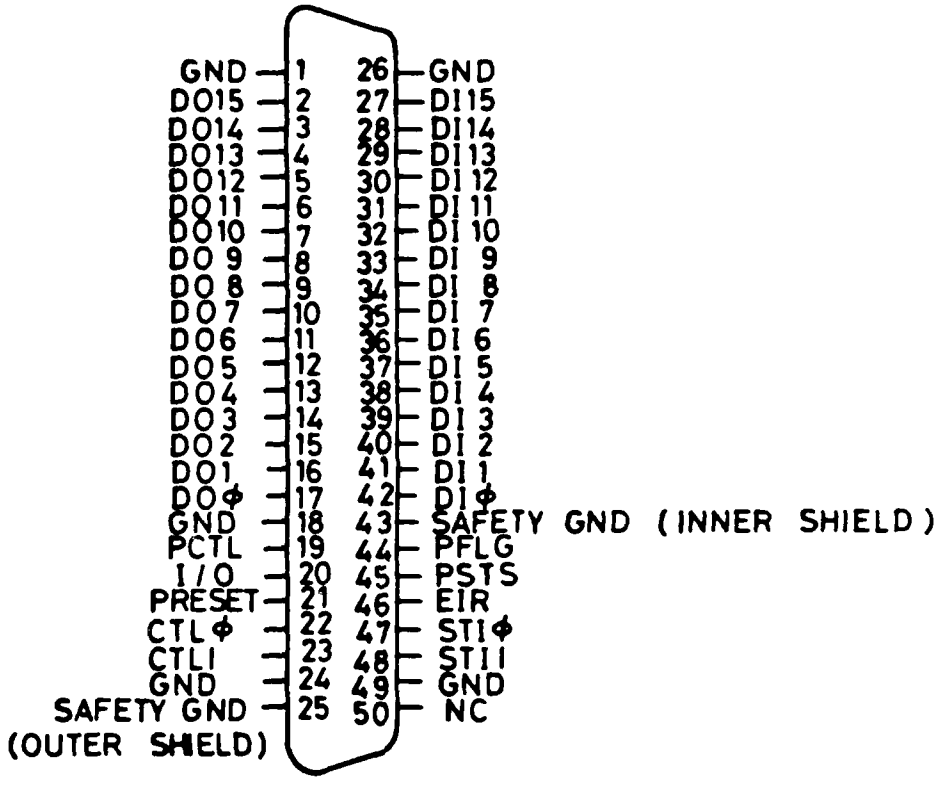


FIG. 5 GPIO CONNECTOR ASSIGNMENT

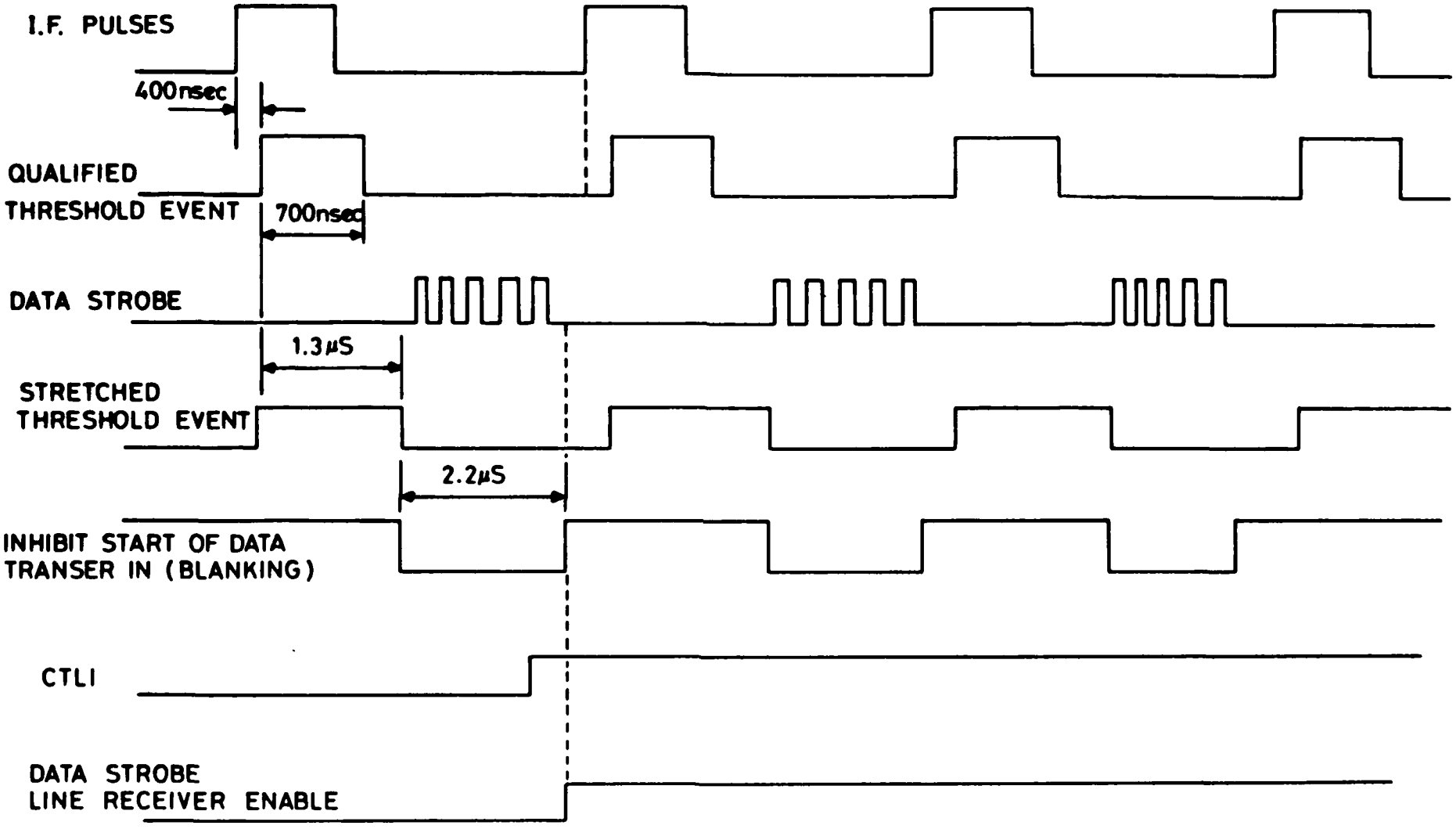


FIG.6 FIFOM INTERFACE TIMING DIAGRAM

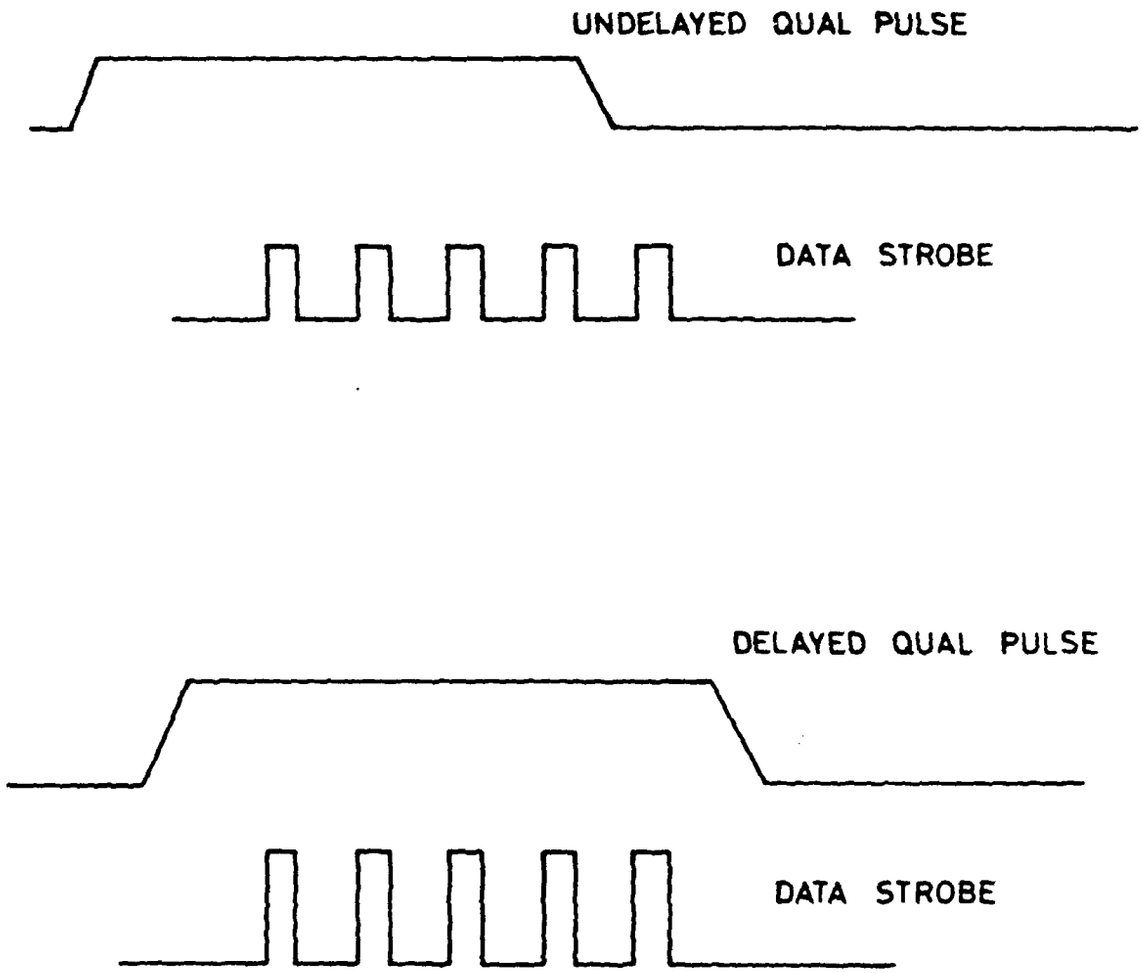


FIG.7 QUAL PULSE TIMING ERROR

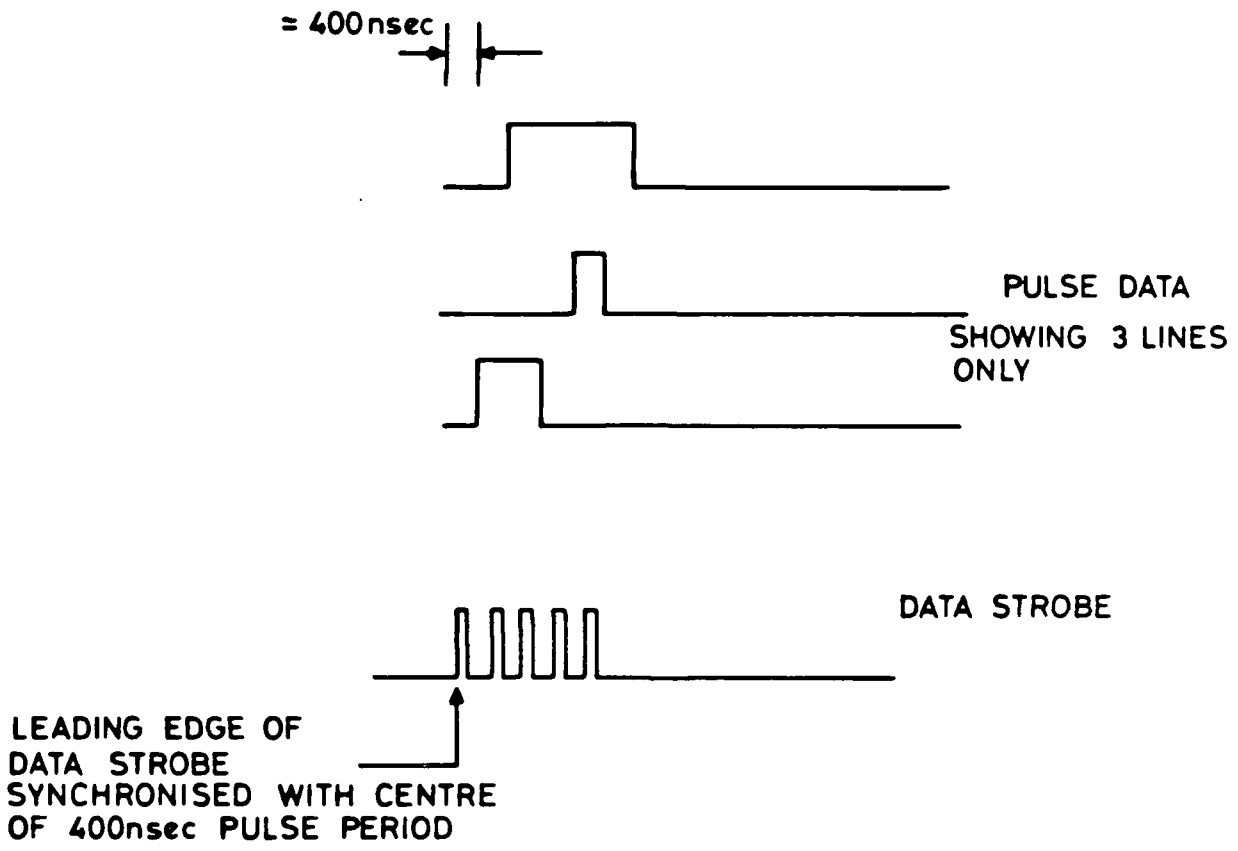
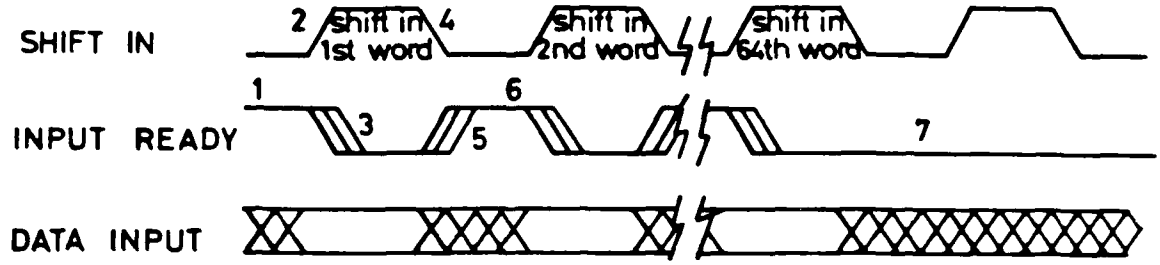
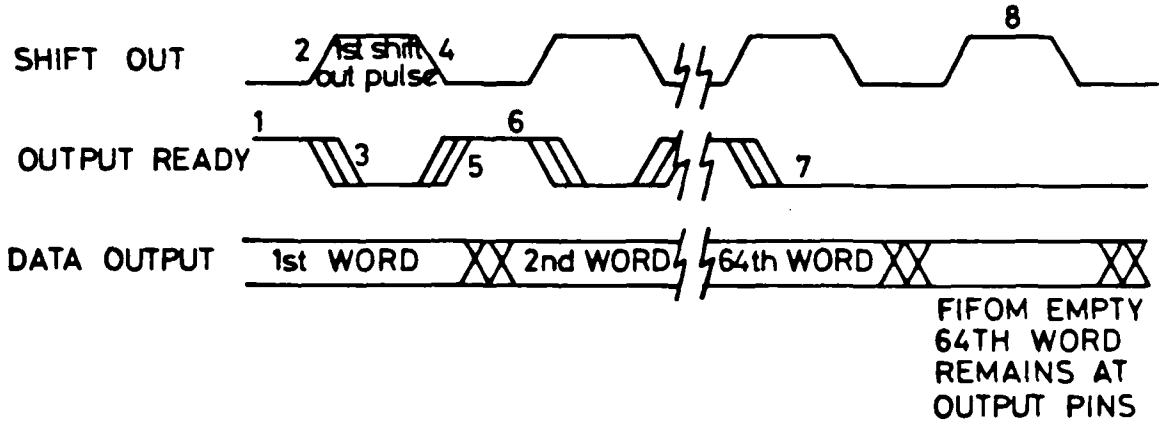


FIG. 8 DATA STROBE / PULSE DATA TIMING ERROR



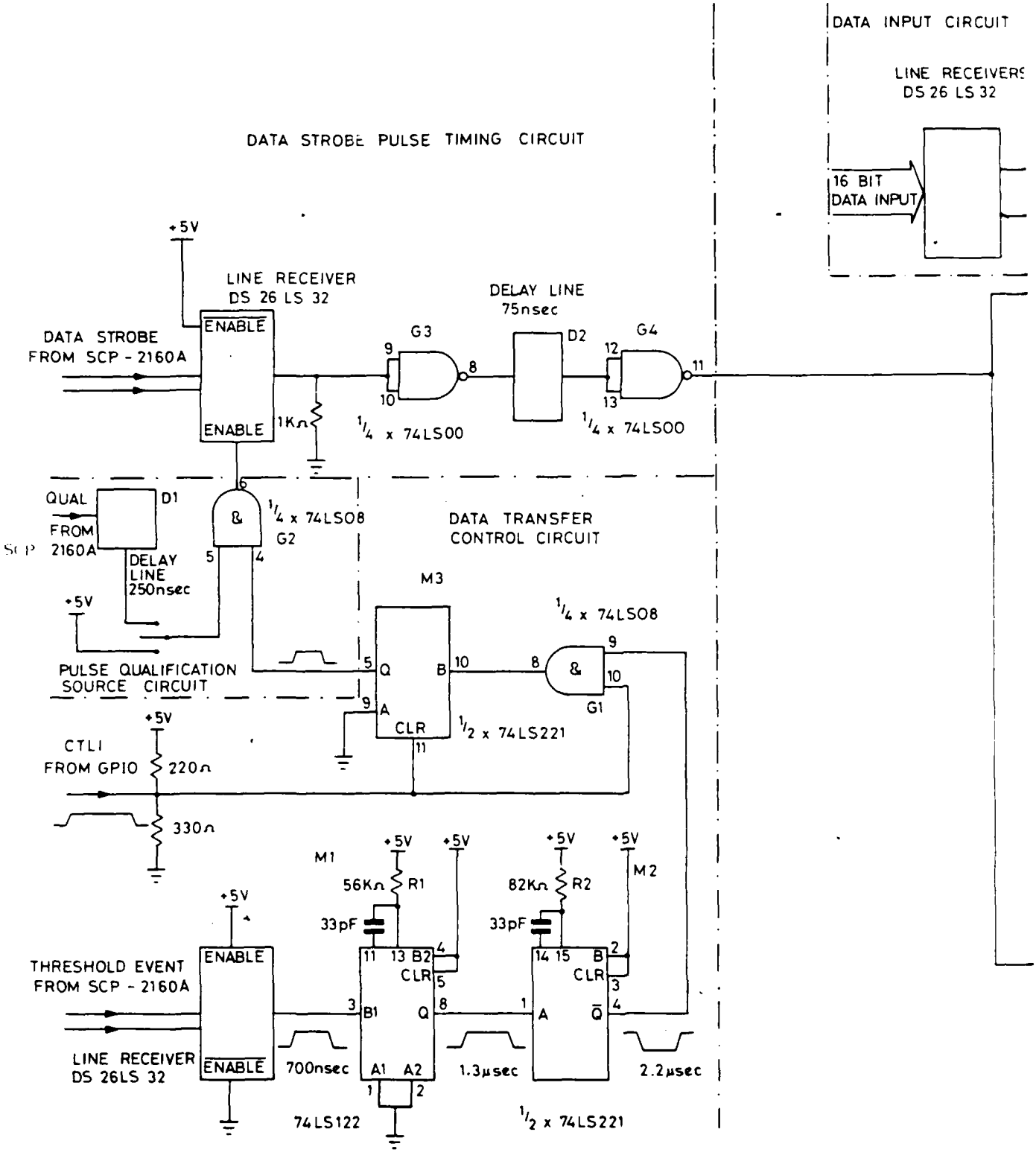
1. Input Ready initially High - FIFOM is prepared for valid data
2. Shift In set HIGH - data loaded into input stage
3. Input Ready drops LOW - input stage "busy"
4. Shift - In set LOW - data from first location falls through memory
5. Input Ready goes HIGH - indicates FIFOM prepared for additional data
6. Repeat process to load 2nd to 64th word into FIFOM
7. Input Ready remains LOW - FIFOM full - no data input occurs

FIG.9 FIFOM SHIFTING - IN SEQUENCE

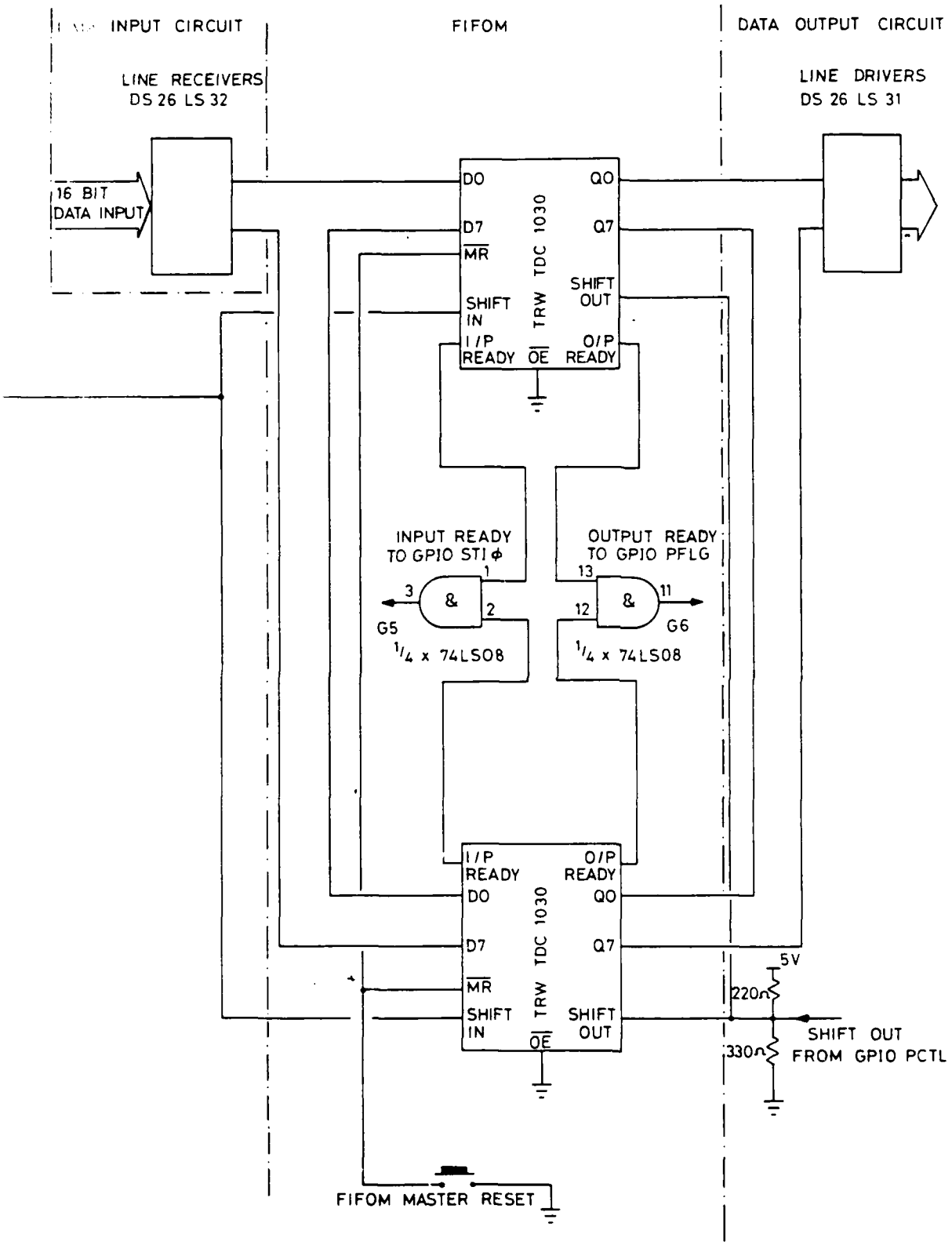


1. Output Ready HIGH - no data transfer in progress, valid data present at output
2. Shift Output set HIGH - results in output Ready LOW
3. Output Ready drops LOW - output stage busy
4. Shift Out set LOW - data in input stage is unloaded and replaced by new data.
5. Output Ready goes HIGH - transfer process completed, valid data at output
6. Repeat process to unload 3rd to 64th words from FIFOM
7. Output Ready remains LOW - FIFOM is empty
8. Shift Out pulse asserted - no transfer occurs from empty FIFOM

FIG.10 FIFOM SHIFTING - OUT SEQUENCE


DATA INPUT CIRCUIT
LINE RECEIVERS
DS 26 LS 32
16 BIT
DATA INPUT
DATA STROBE PULSE TIMING CIRCUIT
+5V
LINE RECEIVER
DS 26 LS 32
DELAY LINE
75nsec
DATA STROBE
FROM SCP - 2160A
ENABLE
ENABLE
G3
G4
D2
1Kn
1/4 x 74LS00
1/4 x 74LS00
QUAL
FROM
SCP 2160A
D1
DELAY
LINE
250nsec
&
1/4 x 74LS08
G2
DATA TRANSFER
CONTROL CIRCUIT
M3
1/4 x 74LS08
PULSE QUALIFICATION
SOURCE CIRCUIT
Q
B
A
CLR
1/2 x 74LS221
G1
CTLI
FROM GPIO
220n
330n
M1
M2
56Kn
R1
82Kn
R2
33pF
33pF
THRESHOLD EVENT
FROM SCP - 2160A
LINE RECEIVER
DS 26LS 32
ENABLE
ENABLE
B1
B2
CLR
A
Q
A1
A2
700nsec
1.3µsec
2.2µsec
74LS122
1/2 x 74LS221

192


INPUT CIRCUIT
FIFOM
DATA OUTPUT CIRCUIT
LINE RECEIVERS
DS 26 LS 32
LINE DRIVERS
DS 26 LS 31
16 BIT
DATA INPUT
DO
D7
MR
SHIFT
IN
I/P
READY
OE
Q0
Q7
SHIFT
OUT
O/P
READY
TRW TDC 1030
INPUT READY
TO GPIO STI ϕ
OUTPUT READY
TO GPIO PFLG
1
2
3
&
G5
1/4 x 74LS08
13
12
11
&
G6
1/4 x 74LS08
5V
220Ω
330Ω
SHIFT OUT
FROM GPIO PCTL
FIFOM MASTER RESET

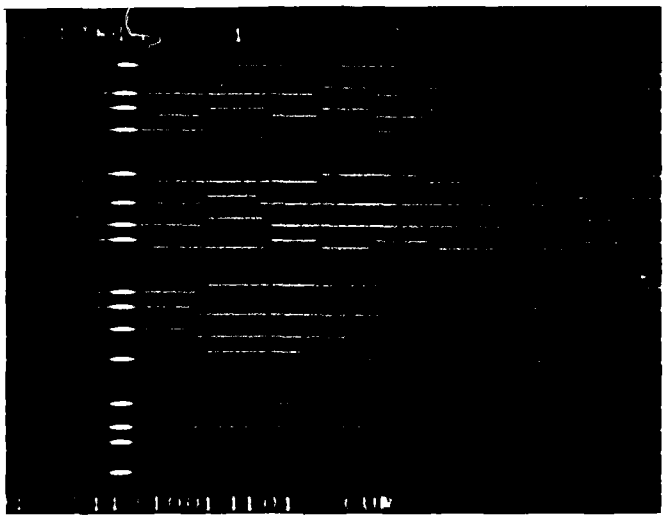


FIGURE 12

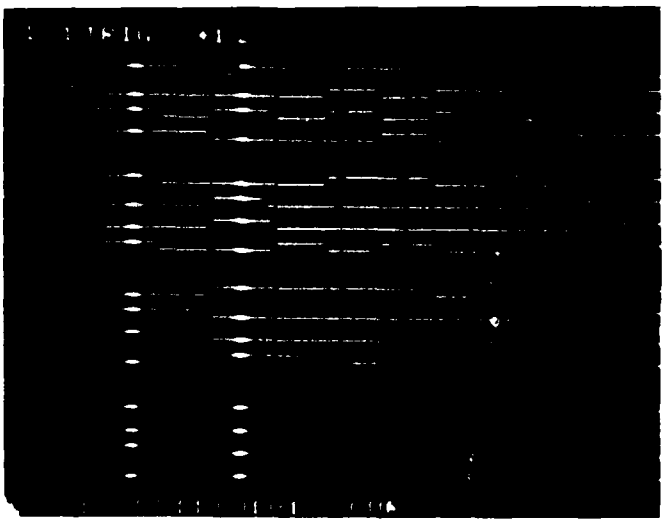


FIGURE 13

DOCUMENT CONTROL SHEET

Overall security classification of sheet ..... UNCLASSIFIED .....

(As far as possible this sheet should contain only unclassified information. If it is necessary to enter classified information, the box concerned must be marked to indicate the classification eg (R) (C) or (S))

1. DRIC Reference (if known)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification
	MEMO 4030		Unlimited

5. Originator's Code (if known)	6. Originator (Corporate Author) Name and Location
778400	RSRE St Andrews Road, Malvern, Worcs. WR14 3PS

5a. Sponsoring Agency's Code (if known)	6a. Sponsoring Agency (Contract Authority) Name and Location

7. Title
INTERFACE FOR THE CONNECTION OF THE SCP-2160A PULSE ANALYSER.

7a. Title in Foreign Language (in the case of translations)

7b. Presented at (for conference papers) Title, place and date of conference

8. Author 1 Surname, initials	9(a) Author 2	9(b) Authors 3,4...	10. Date	pp. ref.
WILLIAMS. C.			1987.10	25

11. Contract Number	12. Period	13. Project	14. Other Reference

15. Distribution statement

Descriptors (or keywords)

continue on separate piece of paper

Abstract

This memorandum describes a digital First-In-First-Out Memory (FIFOM), with control circuitry, which is designed as an interface between the Marlborough Communications SCP-2160A Pulse Analyser and any desk-top computer fitted with a H-P 98622A GPIO Interface. The FIFOM Interface is necessary to accommodate the data transfer-speed differential that exists between the SCP-2160A and GPIO port of the computer.

The SCP-2160A is a 160 MHz I.F. pulse processor that digitises, displays, and stores the pulse parameters that are required for ESM applications.


FILMED
58